Theme

An On-line Monitoring and Control System for Semiconductor Manufacturing Process Utilizing Fuzzy Rule Extraction

Jen-Cheng Chen 박사과정
(Chung Yuan Christian Univ.)
Ming Chang 교수
(Chung Yuan Christian Univ.)

1 Introduction

The recent strides in semiconductor and TFT- LCD technology have spawned enormous application possibilities. The manufacturing processes of both are similar but the substrate materials are different. The R&D results in these two products are compatible and can be applied each other. In general, there are many factors which influence production quality. Most of them can be adjusted by changing the recipe, which are the manufacturing processes parameters of the working machines.

For the past years, engineers devote to find out appropriate and steady processes for on-line real-time controlling recipe. Unfortunately, these processes usually operate with complex and nonlinear reactions, and the process parameters always drifting and varying (Jansen, 1990). Hence development of on-line monitoring and control of parameters become very important for these two industries.

In general, recipe adjustment is a trial and error method. However, the manufacturing process is a very complicated nonlinear system. It is very difficult to find out the relationship between the variation of process parameters and product quality. For this reason, most advanced process systems use the neural network to simulate the manufacturing process model. The neural network can learn the knowledge from the process data automatically. Although neural network performs well in many applications, but it works as a black box that cannot make the decision flow. Hence it is difficult for engineers to confirm the suggestion received from the neural network. In the other hand, the expert system reasons with the logic rules stored in the knowledge base, and it can explain the reason of why it makes the decision. However, it is not easy to construct knowledge base expert system (Jackson, 1999). To acquire the knowledge, it is usually accomplished by a series of lengthy and intensive interviews between a knowledge engineer, who is normally a computer specialist, and a domain expert who is able to articulate his expertise to some degree.

A hybrid intelligence system was developed (Chang et al, 2006) to simulate the plasma enhanced chemical vapor deposition (PECVD) process by combining the expert system with a neural network technique. The expert system is used for on-line inspection of manufacturing process and is set by extracting out the Bolin Logic Rules (Chen et al, 2006; Andrews et al., 1995). The process inputs and outputs of the trained neural network can provide reference to engineers for on-line recipe adjustment. The hybrid intelligence system combines advantages of both neural network and the expert system, which can be learnt from training data and explain the decision flow. The key of the system is rule extraction algorithm. The algorithm is needed to extract the knowledge of the trained neural network model within symbolic rule format, and then store the extracted rules to the knowledge base expert system model. Hence, a rule extracted algorithm called Bound Decomposition Tree (BDT) is designed (Chen et al, 2006). Although the neural network model is developed by using the on-line manufacturing process parameters, the if-then logic rule only provides a qualitative analysis of the deposited membrane. Quantitative prediction of the control of membrane thickness is still a problem.

A novel idea based on fuzzy logic technique is presented here. The ration fuzzy rules between input and output are extracted from neural network with fuzzy rule extraction algorithm. The fuzzy rules extraction method called Fuzzy Bound Decomposition Tree (FBDT) algorithm is presented in section 2, which is a very important part of the hybrid intelligence system. Section 3 portrays verification of the system with the real manufacturing process data collected from the deposition and etching machine respectively.

2. Fuzzy Rule Extraction Method

2.1 Neural Network and Cube of Input Vectors

A neuron is an information processing unit that is the fundamental component of a neural network (Freeman and Skapura, 1992). A neuron has N inputs and can be expressed as $x = \{x_1, x_2, \ldots, x_N\}$, here x_i is usually normalized to a value between 0 and 1. The normalized output of the neuron is $y = f(u(x))$, here $f(\bullet)$ is an specified active function and $f(u) = \text{logsig}(u)$ (Klir and Yuan, 1999). $u(x)$is considered here as $u(x)=\sum_{i=1}^{N} w_i x_i + w_0 = w^T x + w_0$, where $w = \{w_1, w_2, \ldots, w_N\}$ is the weight vector, and w_0 is the bias.

In general, the region of input x_i can be sorted and described with a fuzzy sets $Z_i \in [Z_{i1}, Z_{i2}, \wedge, Z_{im}]$, and each fuzzy set Z_{ij} (i = 1~N, j = 1~m) has a corresponding fuzzy range $[v_{ij}^L, v_{ij}^U]$. The value of x_i must fit in one of the sets, that is $0 \le v_{ij}^L \le x_i \le v_{ij}^U \le 1$. Similarly, the output y of the neuron fits in one set of a fuzzy sets $Z^y \in [Z_1^y, Z_2^y, \wedge, Z_p^y]$, and each fuzzy set Z_k^y (k = 1~p) has a fuzzy range $[vy_k^L, vy_k^U]$ with values of $0 \le vy_k^L \le y \le vy_k^U \le 1$. In addition, Z_k^u is the fuzzy set corresponds to Z_k^y in $u(x)$ domain, and it also has a fuzzy range $[vu_k^L, vu_k^U]$, where $vu_k^L = \text{logsig}^{-1}(vy_k^L)$ and $vu_k^U = \text{logsig}^{-1}(vy_k^U)$.

A cube is a set which contains all input vectors of x in a neuron and can be expressed as cube(w*), where $w^* = \{w_1, w_2, \ldots, w_N, w_0\}$ includes the weights and bias of the neuron. The bound of a cube is the maximum and minimum of $u(x)$ in the cube and can be shown as *bound(cube(w*)) = [Lbound, Ubound]*, where the lower bound *(Lbound)* is the minimum of $u(x)$ in the cube, and is equal to $\sum_{i, w_i<0} w_i + w_0$; the upper bound *(Ubound)* is the maximum of $u(x)$ in the cube, and is equal to $\sum_{i, w_i<0} w_i + w_0$.

It is easy to find that the absolute maximum of

$u(x)$ occurs at $x_i = 1$, for this the weights w_i get positive value and $x_i = 0$ for the weights wi is negative, and the minimum of $u(x)$ occurs at $x_i = 1$ with w_i is negative and $x_i = 0$ with w_i is positive. Accordingly, the upper bound of the cube is the sum of the bias w_0 and all w_i with positive values; and the lower bound of the cube is the sum of w_0 and all w_i with negative values.

In order to extract the fuzzy rules governing a neuron to activate it, the cube of the neuron must be continuously sorted into sub-cubes until the bound values of all sub-cubes lie in between the fuzzy range $[vy_k^L, vy_k^U]$ in a certain fuzzy set Z_y^k. That means all inputs x of the cube will make the output y of the neuron belongs to Z_y^k. In the present research, x_1 is assigned to sort into certain ranges of Z_{11}~Z_{1m}, so that the original cube is divided into m sub-cubes, and each input of x_1 will be assigned to fit in one set of the sets Z_1. Similarly, each resultant sub-cube can be sorted again and divided into smaller sub-cubes by assigning x_2. The smaller subset of the sub-cube of input vectors can be obtained with assignment of more xi terms into sub cubes. A sub-cube of a cube is a subset of the input vectors x in a neuron. When the first k terms of an input vector x_i in a cube are sorted and assigned, a sub-cube which undergoes k times sorting is obtained and can be expressed as $cube(w^*, Z_1Z_2 \dots Z_k)$, where $Z_1Z_2 \dots Z_k$ are the related fuzzy sets of the first assigned xi terms.

It is known that each sub-cube is a sub-set of the input vector, and each sub-cube with different inputs cause different values of u(x). The bound of a sub-cube is the maximum and minimum of u(x) in the sub-cube and can be written as

$bound(cube(w^*, Z_1Z_2 \dots Z_k)) = [Lbound, Ubound]$

where the lower bound (*Lbound*) of the sub-cube is the minimal $u(x)$ in the sub-cube, and the upper bound (*Ubound*) of the sub-cube is the maximal $u(x)$ in the sub-cube, i.e

$$\begin{aligned} & \boldsymbol{Lbound}(\boldsymbol{cube}(\mathbf{w}^*, Z_1Z_2 \dots Z_k)) \\ &= min\{\, u(\mathbf{x}) \mid x \in \boldsymbol{cube}(\mathbf{w}^*, Z_1Z_2 \dots Z_k)\} \\ &= \sum_{i=1, w_j \ge 0}^{k} v_i^L w_i + \sum_{i=1, w_j < 0}^{k} v_i^U w_i + \sum_{j=k+1, w_j < 0}^{n} w_j + w_0 \end{aligned} \tag{1}$$

and

$$\begin{aligned} & \boldsymbol{Ubound}(\boldsymbol{cube}(\mathbf{w}^*, Z_1Z_2 \dots Z_k)) \\ &= max\{\, u(\mathbf{x}) \mid x \in \boldsymbol{cube}(\mathbf{w}^*, Z_1Z_2 \dots Z_k)\} \\ &= \sum_{i=1, w_j \ge 0}^{k} v_i^U w_i + \sum_{i=1, w_j < 0}^{k} v_i^L w_i + \sum_{j=k+1, w_j \ge 0}^{n} w_j + w_0 \end{aligned} \tag{2}$$

The calculations of bounds via above equations are tedious. Since the bounds of a cube is obtained before the cube is divided into sub-cubes, a more effective method to find out the new bounds of the sub-cube can be achieved with a simpler calculation from the original bounds of the cube. For any cube $cube(w^*, Z_1Z_2 \dots Z_k)$ with bounds of $[Lboundk, Uboundk]$, the bounds of the new sub-cube $cube(w^*, Z_1Z_2 \dots Z_kZ_{k+1})$ with x_{k+1} sort can be obtained from

$$[Lbound_{k+1}, Ubound_{k+1}] = \begin{cases} [Lbound_k + v_{k+1}^L * w_{k+1}, Ubound_k - (1 - v_{k+1}^U) * w_{k+1}], & w_{k+1} \ge 0 \\ [Lbound_k - (1 - v_{k+1}^U) * w_{k+1}, Ubound_k + v_{k+1}^L * w_{k+1}], & w_{k+1} < 0 \end{cases} \tag{3}$$

The above equations are simpler and hence the values of the bounds of the subcube real time can be obtained. Through the calculated values of the bounds, some special cubes are defined as follows: A sub-cube is called a certain cube if it's *Lbound* > vu_k^L and *Ubound* < vu_k^U. That means the output of the neuron will always fit in the fuzzy range of the set $Z^y{}_k$ for all inputs of the cube; otherwise, a sub-cube is an uncertain cube if it is not a certain cube.

Since a certain cube's bound range always fit in the fuzzy range $[vu_k^L, vu_k^U]$ of $Z^y{}_k$, the fuzzy rule to decide the output of a certain cube $cube(w^*, Z_1Z_2 \dots Z_k)$ can be described as "IF x_1 is assigned in Z_1 AND

x_2 assigned in Z_2 AND ... AND x_k assigned in Z_k then the output of the neuron falls in the range of Zy_k".

2.2 Fuzzy Bound Decomposition Tree Algorithm (FBDT)

Input : A neuron's weights and bias

Output : Extracted fuzzy rules of the neuron

Step 1. Set *certain cube sets* $c_1 \sim c_p$ as empty
Set *uncertain cube set* as *cube(w*)*
Set *rule sets* $r_1 \sim r_p$ as empty

Step 2. Select a $cube(w^*)_a$ from uncertain cube set

Step 3. Divide the $cube(w^*)_a$ into m sub-cubes : $cube(w^*)_{a1} \sim cube(w^*)_{am}$

Step 4. Calculate the bound of $cube(w^*)_{a1} \sim cube(w^*)_{am}$ with Eq. (3)

Step 5. Check $cube(w^*)_{a1} \sim cube(w^*)_{am}$ separately,
If the cube is a certain cube of Zy_k, add it to certain cube set c_k
Else add it to uncertain cube set

Step 6. If the uncertain cube set is not empty go to Step 2.

Step 7. For each certain cube sets c_k
Transform each cube in c_k into a rule and insert it into the rule set r_k.

Step 8. Then, each rule set r_k contains the rules which make the neuron's output fit in the range of Zy_k.

2.3 Trust Value of Extracted Rules

With the above algorithm, it can extract rules from a trained neural network. But it does not show the importance of each rule. So, it needs to find out the trust value of each extracted rule, and the basic idea is that if the rule merges with the training data, then it is trustable and important. The hamming distance of a rule and training data can show how close they are. The hamming distance HM of a rule $R=[Z^R_1 Z^R_2 ... Z^R n]$ and a data $D=[Z^D_1 Z^D_2 ... Z^D n]$ is

$$HM = \sum_{i=1}^{n} h_i \quad \text{where} \quad h_i = \begin{cases} 0 & , Z_i^R = 0 \\ abs\left(Z_i^R - Z_i^D\right) & , Z_i^R \neq 0 \end{cases} \tag{4}$$

Then each rule and training data can be checked, if their hamming distance is close enough (smaller) and it can raise the rule's trust value. Thus the trust value T of an extracted rule is

$$T = \sum_{i=1}^{m} t_m, \quad t_m = \begin{cases} 0 & , HM > K \\ K - HM & , HM \leq K \end{cases} \tag{5}$$

where m is the amount of training data and K is a constant. If the hamming distance of a rule and data is smaller than K (close enough), the trust value of the rule will be increased.

3. Examples

Two examples are discussed here where the manufacturing process data are collected from etching and deposition machine. These data are used to check the intelligence system and the results are illustrated in the following subsections.

3.1 Deposition Manufacturing Process

3.1.1 The Training Data

The training data of the neural network are collected from the CVDs' quality control (QC) department of ChungHwa Picture Tubes (CPT). 1425 sets of records of raw data are collected from PECVD machines, where 1000 sets of them are used as the training data of the neural network. These raw records contain 8 kinds of parameters, e.g.

x_1. IN Temperature
x_2. OUT Temperature
x_3. RF Power
x_4. Pressure

x_5. Flow of Gas NH3
x_6. Flow of Gas SiH4
x_7. Flow of Gas N2
x_8. Reflect Power

These parameters are used as the inputs of the training data, and the four values of membrane thickness including average, maximum, minimum, and uniformity are chosen as outputs of the training data.

3.1.2 Establishment of Neural Network Model

As mentioned above, a neural network with 8 inputs and 4 outputs is developed and studied with 1000 sets training data. The neural network model is generated with back-propagation network. After training with those 1000 sets of training data, another 425 inputs are used as test data to check the precision of the trained neural network. Average error for the average, maximum and minimum of the predicted membrane thickness is shown in Fig.1. It is observed that the predicted values are ± 12.97 %, ±8.95 %, and ±10.97 %, corresponding to the real values of membrane thickness ±38.9 Å, ± 67.15 Å, and ±32.92 Å, respectively. It can also predict the uniformity of the membrane where the test error is ±6.96 %. Since all of the predicted accuracies are approximately 90 %, this model can be used for the following procedure of rule extraction.

3.1.3 Extracted Rules from the Trained Network

After the neural network has been trained with the training data, the fuzzy rule is used to monitor and control of manufacturing process. The process parameters are then extracted from the trained neural network. Here each input is separated into 6 fuzzy zones [Z_{i1}, Z_{i2}, ..., Z_{i6}] according to the raw data separately, and each output is separated into 5 fuzzy zones [Z_{y1}, Z_{y2}, ..., Z_{y5}] according to the measured membrane thickness. Using the above FBDT algorithm, the difference in rules from each output and fuzzy zones can be extracted. The numbers of extracted rules from each output is

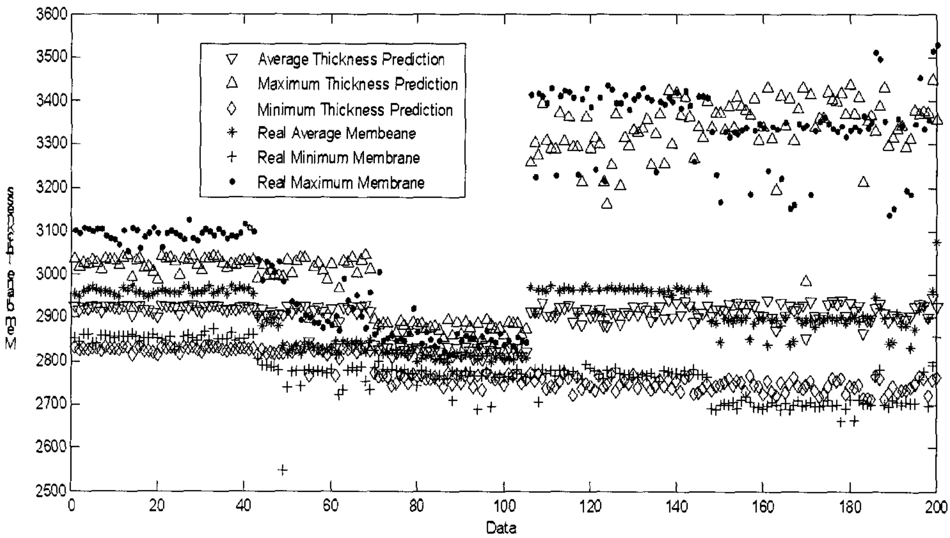

Fig. 1. The predicted results for maximun, minum, and average value of the membrane thckness.

shown in Table 1. Although the amounts of the extracted rules are large, the time for making decision can be completed within few seconds via a computer. Besides, we calculate the trust value of each extracted rule using Eq. (4) & Eq. (5). The expert system can infer the rules with the order of trust value, so that it can increase the reason efficiency which arise the trust of the expert system's reason result. Based on these extracted fuzzy rules, an expert system for on-line inspection and control of process parameters can be designed. For example, Table 2 shows the eight extracted rules for the output of average of membrane thickness in the zone of Zy_3 which has highest trust value. The number in the table is the fuzzy zone for each input and 0 means the state of "don't care". Accordingly, rule R1 indicates:

" IF x_1 is in the 4th fuzzy state Z_{14} AND x_3 in the 1st fuzzy state Z_{31}
AND x_4 in the 5th fuzzy state Z_{45} AND x_5 in the 4th fuzzy state Z_{54}
AND x_6 in the 4th fuzzy state Z_{64} AND x_8 in the 1st fuzzy state Z_{81}
THEN the average of membrane thickness will be in the 3rd fuzzy state of Zy_3".

That is

" IF x_1 (IN Temperature) is between [282.5 283.5]
AND x_3 (RF Power) is between [5700 5740]
AND x_4 (Pressure) is between [1790 1810] AND x_5 (Flow of Gas NH3) is between [4300 4386]
AND x_6 (Flow of Gas SiH4) is between [1340 1355] AND x_8 (Reflect Power) is between [0 5]
THEN maximum membrane thickness will be between 2839~2918Å".

Table 1. Numbers of extracted rules for each output.

	Avg	Max	Min	U
Zy_1	2061	9250	1267	12521
Zy_2	4119	13224	4678	16154
Zy_3	5508	14542	9019	14930
Zy_4	6398	14915	10966	19422
Zy_5	2865	6935	4001	11739

Table 2. First eight extracted rules for the control of average of membrane thickness in the Zy_3 state with biggest trust values.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Trust Value
R1	4	0	1	5	4	4	0	1	4833
R2	3	0	1	5	4	4	0	1	4708
R3	4	0	2	4	4	4	0	1	4378
R4	3	0	2	4	4	4	3	1	4167
R5	5	0	1	5	4	4	0	1	3994
R6	4	0	1	5	4	3	0	1	3981
R7	2	0	1	5	4	4	0	1	3974
R8	3	0	1	5	4	5	0	1	3919

3.1.4 Accuracy of the Expert System

Based on these extracted rules, a quantitative prediction of the membrane thickness is executed and compared to the true data. The error rates obtained for each output is shown in Table 3. It shows that the current errors are from 10 % to 15 %. This may be caused by the tilt of machine substrate which can be resolved by adding a virtual parameter into the system. Besides, some of the training data are concentrated in a certain range; which is not satisfactory for the generation of the training model. This could be improved in the stage of data collection.

Table 3. Prediction errors based on extracted rules.

	Average	Maximum	Minimun	Uniformity
Errors	12.18 %	15.02 %	13.88 %	10.72 %

3.2 Etching Process

The training data of the neural network are collected from an ECR Etcher of DRAM industry. 99 sets of raw data are collected from the process chamber in the ECR etcher machine. These original raw data contain 8 parameters which are used as the inputs of the training data for neural network are:

x_1. Process time (Step Time+over etching time, sec)
x_2. Gas1(sccm) (Ar)
x_3. Pressure(Pa)
x_4. Mag RF Pf (w)
x_5. Mag RF Pr(w)
x_6. Backside Pressure(Kpa)
x_7. Circulator(℃)
x_8. Block Temp(℃)

70 sets of raw data are used as the training data of the neural network, and fours values of etching depth including average, maximum, minimum, and standard deviation are chosen as the outputs of the training data.

3.2.1 Establishment of Neural Network Model

As mentioned, 70 sets of raw data are used as the training data, and the other 29 inputs are used as test data. Figure 2 shows the average errors obtained for the maximum, minimum and average of the predicted etching depth are ±14.32 %, ±12.66 %, and ±12.62 %, corresponding to the real values of depth ±78.62 Å, ±71.66 Å, and ±68.65 Å, respectively. Since the predicted accuracies are approximately 86 %, this model is acceptable to be used in the rule extraction for etching process.

3.2.2 Extracted Rules from the Trained Network

Similarly, each input was separated into 6 fuzzy zones [Z_{i1}, Z_{i2}, ..., Z_{i6}], and each output into 5 fuzzy zones [Zy_1, Zy_2, ..., Zy_5], the process parameters are then extracted from the trained neural network. The obtained numbers of extracted rules for each output are shown in Table 4. Again, they can be reduced

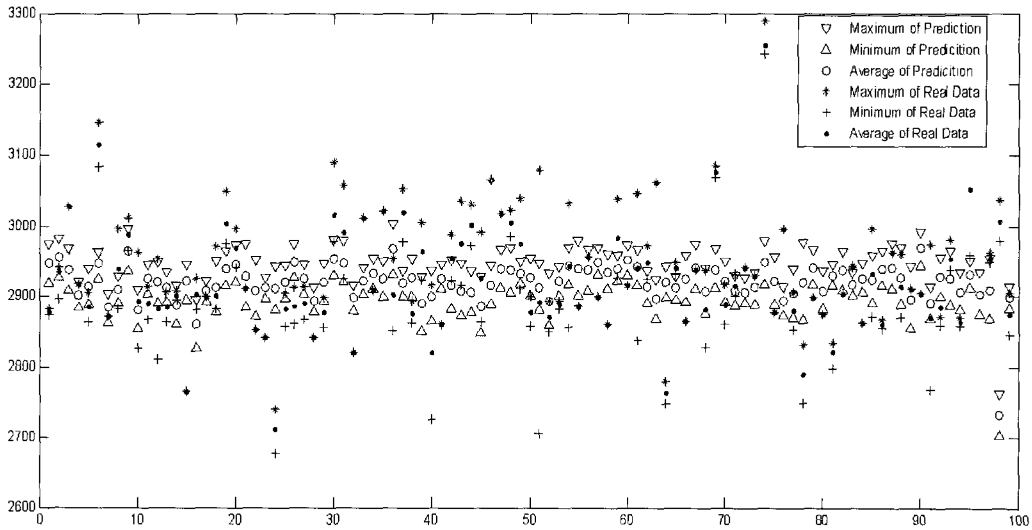

Fig. 2. The predicted results for maximun, minum, and average value of etching depth.

with the thresholds of the trust value. Table 4 shows the eight rules of the average of etching depth in the Zy_3 state which has highest trust value. Based on these extracted fuzzy rules, an expert system for on-line inspection and control of process parameters can be expected.

Table 4. First eight extracted rules for the control of average of etching depth in the Zy_3 state.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Trust value
R1	6	0	0	3	0	0	1	2	578
R2	6	0	0	2	0	0	1	3	566
R3	6	0	0	2	0	0	1	2	543
R4	6	0	0	3	0	0	2	3	532
R5	6	0	0	3	1	0	1	3	515
R6	6	0	0	3	0	0	2	2	511
R7	6	0	0	2	0	0	2	3	497
R8	6	0	0	3	2	0	1	3	496

From Table 4, rule R1 indicates

"IF x_1 is in the 6th fuzzy state Z_{16}
AND x_4 in the 3rd fuzzy state Z_{43}
AND x_7 in the 1st fuzzy state Z_{71}
AND x_8 in the 2nd fuzzy state Z_{82}
THEN the average of etching deep will be in the 3rd fuzzy state of Zy_3".

That is

"IF x_1 (Process time) is between [61.78 63.1]
AND x_4 (Mag Pf) is between [352.1 352.5]
AND x_7 (Circulator) is between [-20.2 -19.91]
AND x_8 (Block Temp) is between [99.41 99.83]
THEN the average etching deep will be between 2929.6~3038.4".

3.2.3 Accuracy of the System

The prediction of the etching depth based on these obtained extracted rules is executed and compared with the neural network prediction and true data. The resultant error rates for each output are shown in Table 5. Similarly, it shows that the current errors compare to true data are also roughly from 10 % to 15 %.

Table 5. Etching depth prediction errors.

Errors	Maximum	Minimum	Average	STD
Compare with NN prediction	8.3 %	7.7 %	8.3 %	1.8 %
Compare with true data	14.75 %	14.34 %	13.33 %	10.3 %

4. Conclusions

A fuzzy rule extraction algorithm based on neural networks for the inspection and control of manufacturing process in semiconductor and TFT-LCD industries is presented. Two manufacturing processes including deposition and etching were operated and discussed. The current prediction accuracy for membrane thickness and etching depth is approximately 90 %. Therefore, the constructed knowledge base system can provide a reference to the engineers for recipe adjustments. At present the manufacturing process control work must be executed with the help of a computer since the extracted fuzzy rules are too complicated. A more effective condensed method will be developed in future to reduce the number of rules, then the need of real-time prediction and control of manufacturing process could be expected.

References

[1] Andrews, R., Diederich, J., and Tickle, A. B., "Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks," Knowledge Base Systems, Vol. 8, No. 6, pp. 373-389 (1995).

[2] Chang, M., Chen, J. C., Chang, J. W., and Heh, J. S., "Advanced Process Control Expert System of CVD

Membrane Thickness Based on Neural Network," Material Science Forum, Vol.505-507, pp. 313-318 (2006).
[3] Chang, M., Chen, J. C., Chang, and Heh, J. S., "The Control of Membrane Thickness in PECVD Process Utilizing a Rule Extraction Technique of Neural Networks," Lecture Notes in Computer Science, Vol.3973, pp. 1091-1098 (2006).
[4] Chen, J. C., Heh, J. S., and Chang, M., "Designing a Decompositional Rule Extraction Algorithm for Neural Networks," Lecture Notes in Computer Science, Vol. 3971, pp. 1305-1311 (2006).
[5] Chen, J. C., Chang, M., and Heh, J. S., "An Advanced Manufacturing Process Control system for PECVD Process Utilizing Fuzzy Logic Technique", Journal of the Chinese Society of Mechanical Engineers (2006b) (accepted)
[6] Freeman, J. A., and Skapura, D. M., Neural Networks, Addison-Wesley, New York, pp.89-126 (1992).
[7] Haykin, S., Neural Netwroks, Prentice Hall, New Jersey, pp.15-34 (1999).
[8] Jackson, P., Introduction to Expert System, Addison-Wesley, New York (1999)
[9] Jansen, F., AVS Short Course: PECVD, American Vacuum Society (1990).
[10] Klir, G., and Yuan, B., Fuzzy Sets and Fuzzy Logic, Prentice Hall, New Jersey, pp.379-417 (1999).
[11] Semiconductor Industry Association, The International Technology Roadmap for Semiconductor: 2001 Edition.
[12] Zadeh, L. A., "Fuzzy logic and approximate reasoning," Synthese, Vol.30, pp.407-428 (1975).

저|자|약|력

성 명 : Jen-Cheng Chen

◈ 학 력
· 1996년
Chung Yuan Christian Univ.(대만) 전자공학과 공학사
· 2000년
Chung Yuan Christian Univ.(대만) 컴퓨터 및 정보공학과 공학석사
· 현 재
Chung Yuan Christian Univ.(대만) 전자공학과 박사과정

성 명 : Ming Chang

◈ 학 력
· 1976년
National Central Univ.(대만) 토목공학과 공학사
· 1980년
National Taiwan Univ. (대만) 토목공학과 공학석사
· 1986년
National Taiwan Univ. (대만) 기계공학과 공학박사

◈ 경 력
· 1980년 - 1982년
Ching-Ling Industrial Research Institute (대만), 연구원
· 1982년 - 1986년
Chung Yuan Christian Univ. (대만) 기계공학과 강사
· 1994년 - 현 재
current, Chung Yuan Christian Univ. (대만) 기계공학과 교수